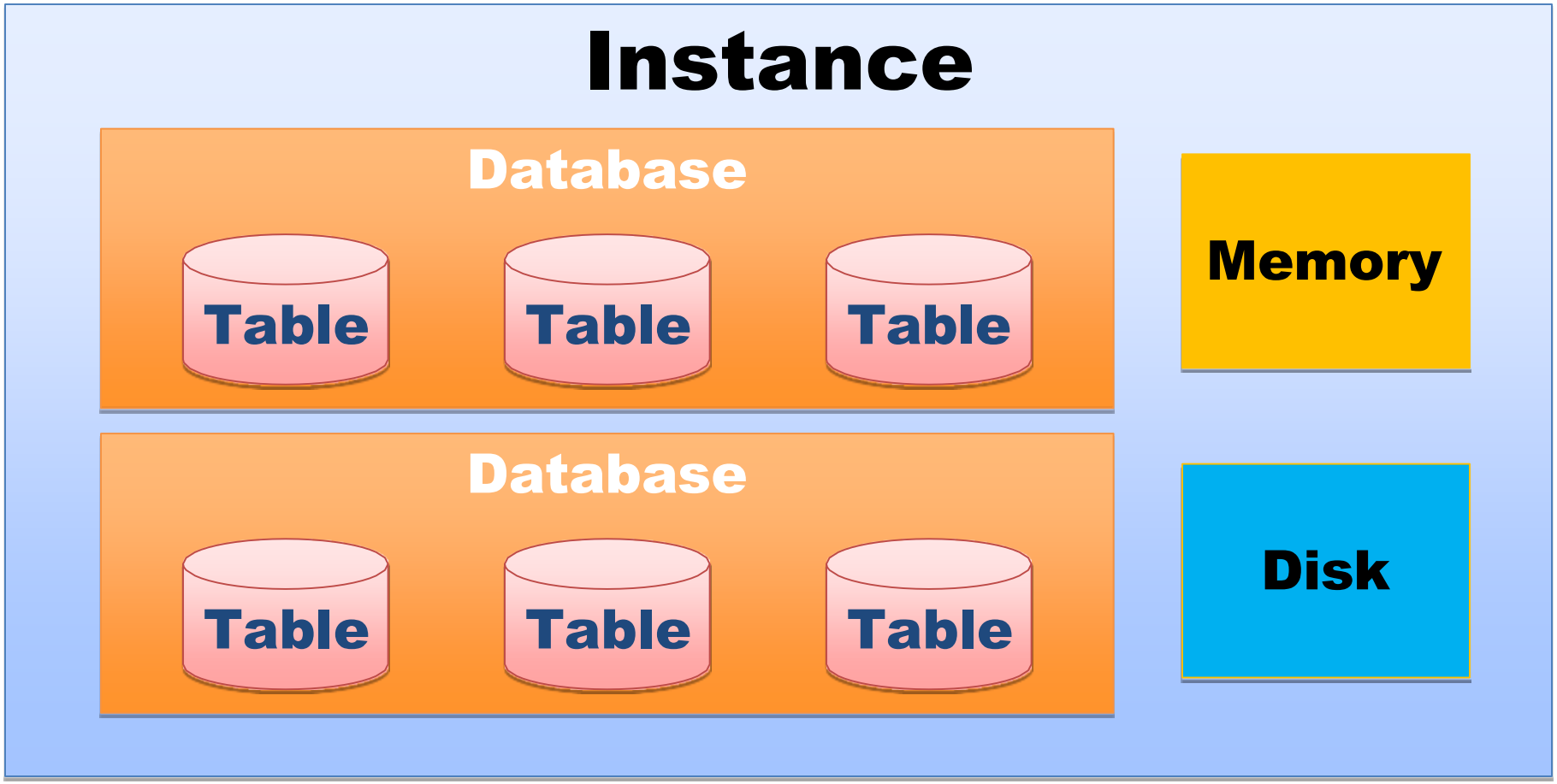
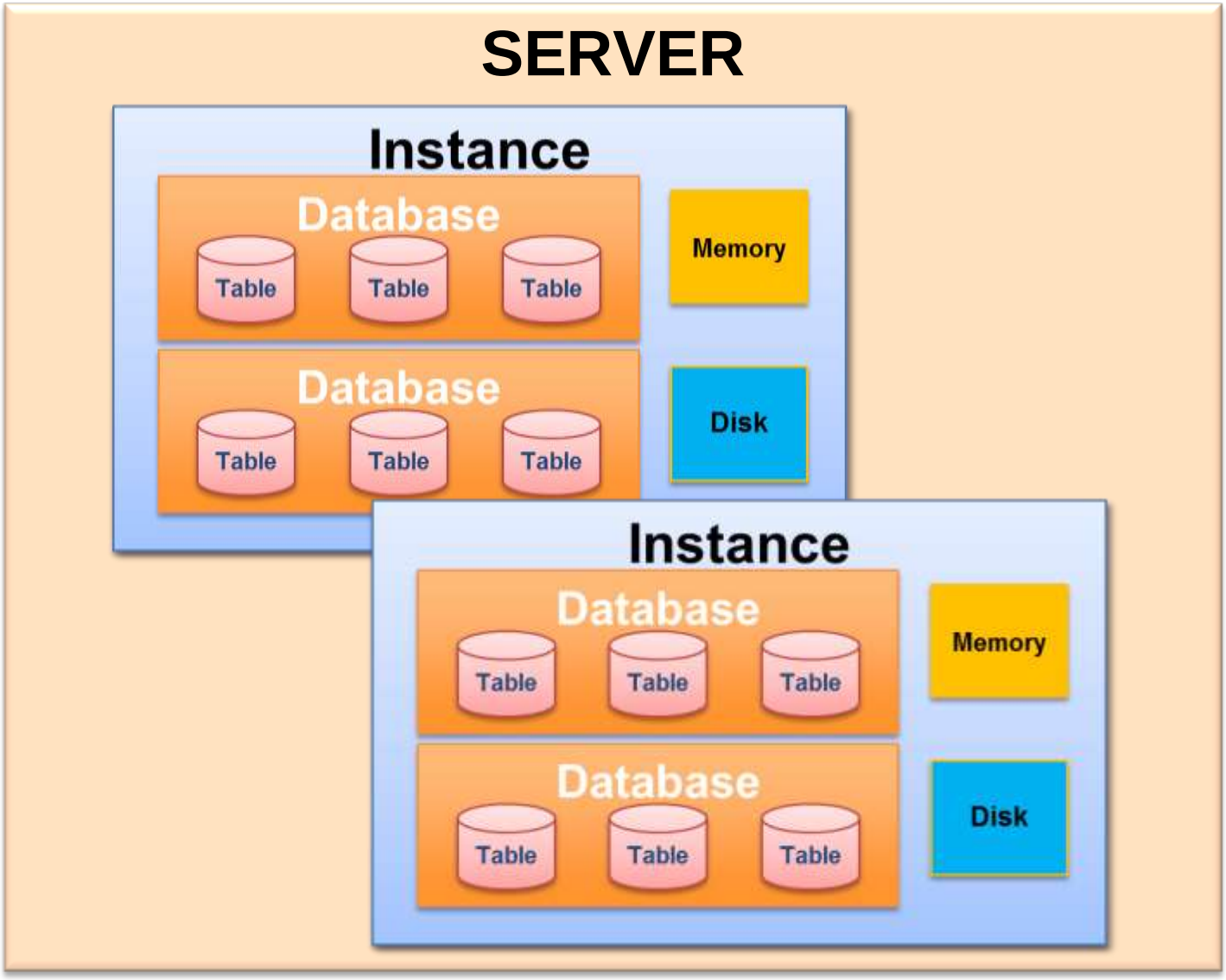
Informix Terms

- **Instance** – An individual copy of Informix running on a server, each must have its own unique ID
- **Database** – Container of tables, procedures, and other elements in an instance, owned by a particular user
- **Table** – Container of data living in a database. By default owned by the database owner



An ***instance*** is a set of resources (disk and memory) that can be shared by multiple ***databases***

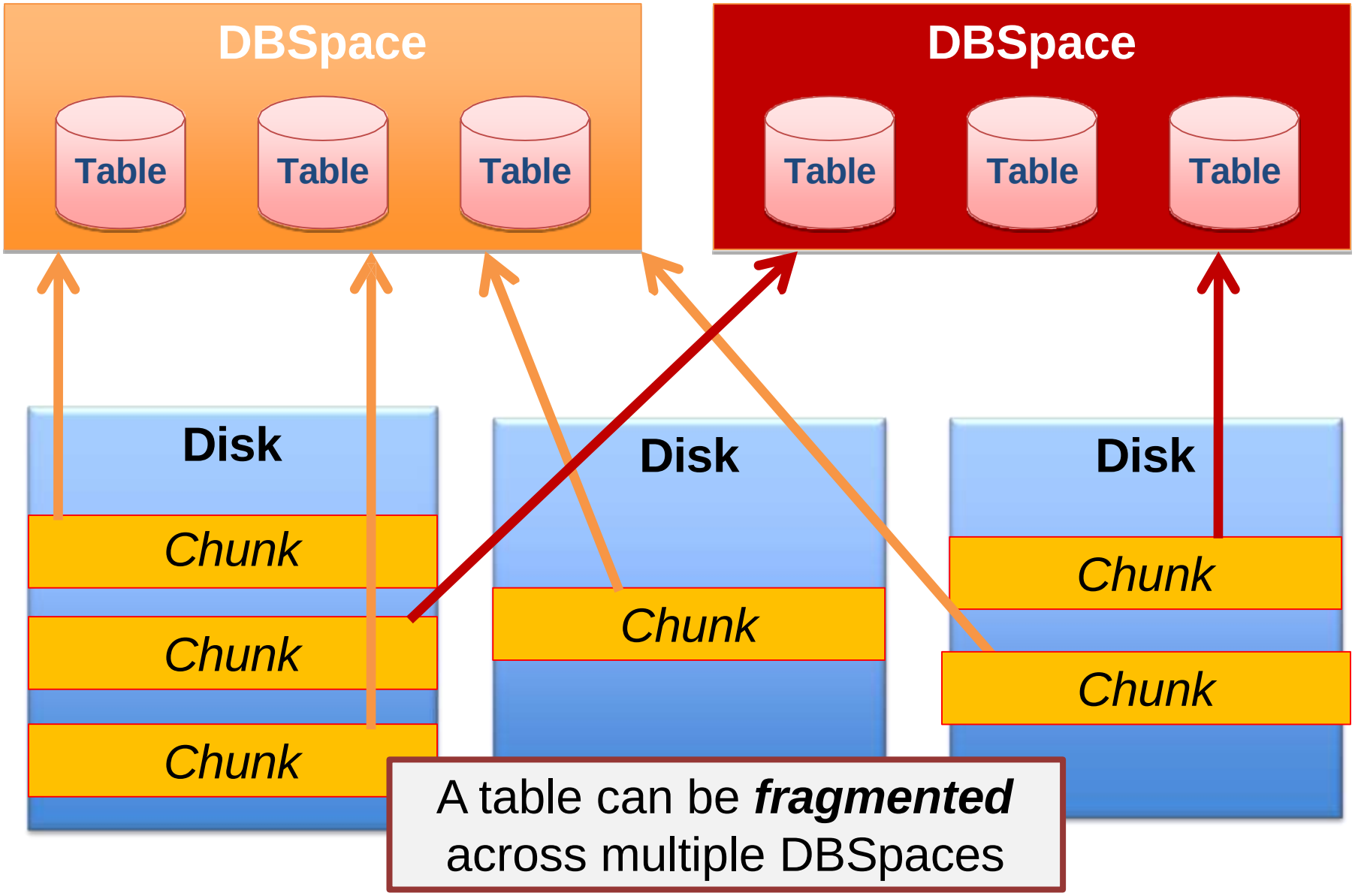


Multiple instances can run on a server, each with their own set of resources

Space Terms

- **DBSpace** – Set of one or more chunks that store Informix data
- **Chunk** – A file or device pre-allocated for space that make up a DBSpace
- **Bufferpool** – The buffers in memory that hold retrieved data for faster processing

DBSpace/Chunks



Informix Commands

oninit	central server application
onmode	manages the running engine
onstat	gives statistics and information on the current instance
ontape	a backup utility
onspaces	creates/deletes chunks
onparams	manages logical and physical logs

Key ONCONFIG Parameters

Will discuss just a few of the many settings…

ROOTPATH	Path to the root DBSpace
ROOTSIZE	Size of the first rootdbs chunk
ROOTOFFSET	Offset on disk, set 0 if using cooked files
LOGFILES	Don't touch this - it will be dynamically adjusted
MSGPATH	Path to server log
DBSPACETEMP	Name of the default temp DBSpace

Key ONCONFIG Parameters

DBSERVERNAME	Shared memory instance name 'test1'
DBSERVERALIASES	Name(s) of aliases, use at least the tcp port name from sqlhosts (*test1_tcp*)
FULL_DISK_INIT	If you need to re-initalize the instance to a factory default, set it to 1

Key ONCONFIG Parameters

SHMVIRTSIZE	Size in KB given to the engine at start time
SHMADD	New memory block size.
SHMTOTAL	Maximum size of memory to use, 0=unlimited
TAPEDEV	Device/File/Directory ontape uses to backup - /dev/null for fake backups
LTAPEDEV	Device/File/Directory ontape uses to backup logical logs - /dev/null to discard the logical logs

BUFFERPOOL

- Parameter(s) in the ONCONFIG file specifying the amount of memory allocated to a cache of data
- Need one bufferpool for tables using the default page size (2 KB or 4 KB depending on the OS)
- Add lines for each additional bufferpool for tables with other page sizes

BUFFERPOOL

- Default line is the template for new bufferpools
  - Version 12:

    `BUFFERPOOL size=page_size,memory=memory_size`

  - Earlier Versions:

    `BUFFERPOOL size=4k,buffers=10000,lrus=8,lru_min_dirty=50,lru_max_dirty=60`

Size	Page size buffer, check version info
Buffers	Number of buffers of page size
LRUs	Number of queues to handle buffers
LRU min/max	Threshold of dirty pages for when to start writing out pages to disk between checkpoints and when to stop

Allocating Space

- Create a directory to store your dbspaces or links to them
- Change owner of directory to informix:informix
- Change the permission to 770
- Create files for storage (these will become the chunks)
- Change owner of files to informix:informix
- Change the permission to 660

Root DBSpace

The root DBSpace is a critical storage space

Example of creating the root space chunk:

```
mkdir /informixchunks
chown informix:informix /informixchunks
chmod 770 /informixchunks
touch /informixchunks/rootdbs
chown informix:informix /informixchunks/rootdbs
chmod 660 /informixchunks/rootdbs
```

This path and filename must match the value of ***ROOTPATH*** *in the* ***ONCONFIG***

Get Going!

```
oninit –iv
```

oninit	The control process
-i	Initalizes a new instance
-v	Verbose
-y	Respond yes automatically

Messages will be displayed during startup. Look for:

```
Verbose output complete: mode = 5
```

Get Going!

```
onstat -

   IBM Informix Dynamic Server
   Version 12.10.FC4 -- On-Line --
   Up 00:00:50 -- 1182476 Kbytes
```

The Informix instance is now up and running!

Allocating Space

- Follow the procedure earlier to create more files as chunks for additional DBSpaces
- Create files for the chunks
  - tmpdbs
  - logdbs
  - datadbs

Creating Chunk Files - Example

```
cd /informixchunks
touch logdbs
touch tmpdbs
touch datadbs
chmod 660 *dbs
chown informix:informix *dbs
```

onspaces

- Now that the instance is online, and files have been created, add additional DBSpaces with the **onspaces** command

onspaces

onspaces

-c	create
-d <*dbspace*>	dbspace name
-s <*size*>	size in bytes
-o <*offset*>	offset
-p <*path*>	full path to file

The offset is used if using a raw device.
For cooked files it generally should be 0

Create DBSpaces - Example

```
onspaces -c -d logdbs -o 0 -s 200000 -p
  /informixchunks/logdbs
```

-t option to specifies a temporary DBSpace

```
onspaces -c -d tmpdbs -t -o 0 -s 200000
  -p /informixchunks/tmpdbs
```

```
onspaces -c -d datadbs -o 0 -s 500000 -
  p /informixchunks/datadbs
```

Restart the Instance

- A restart is sometimes required to make some configuration changes active
- Shutdown the running instance:

```
onmode -kuy
```

- Startup the instance:

```
oninit -v
```

Create a Sample Database

- As user informix, execute '**dbaccessdemo**'
- Creates the *stores_demo* database
- When prompted, answer 'N' to installing sample scripts
  - Copies a series of C scripts to the current directory

Connect To The Database

dbaccess

Curses based tool for executing SQL and simple administration

–Select '**Database**'

–Select '**Select**'

–Choose '**stores_demo@test1**'

- Opens the stores_demo database

–'**Exit**' out of that menu

Connect To The Database

- '**Query-language**'
- '**New**'
- Type:

```
select * from items
```

- Hit '**Esc**' *[Done Editing]*
- '**Run**'
- Browse the data
- Choose '**Exit**' to back out of the menus, then the tool

Add Logical Logs

Add additional logical logs to store transactional information

onparams

-a	add
-d <*dbspace*>	dbspace name
-s <*size*>	size in KB

```
onparams -a -d logdbs -s 50000
```